The 6th Intl. Workshop on Spoken Language Technologies for Under-Resourced Languages
29-31 August 2018, Gurugram, India

A Human Quality Text to Speech System for Sinhala

Lakshika Nanayakkara[1], *Chamila Liyanage*[2],*Pubudu Tharaka Viswakula*[3],*Thilini Nadungodage*[4],*Randil Pushpananda*[5],*Ruvan Weerasinghe*[6]

Language Technology Research Laboratory, University of Colombo School of Computing,Colombo, Sri Lanka

{aln[1],cml[2],hnd[4],rpn[5],arw[6]}@ucsc.cmb.ac.lk,striderpaw@gmail.com[3]

Abstract

This paper proposes an approach on implementing a Text to Speech system for Sinhala language using MaryTTS framework. In this project, a set of rules for mapping text to sound were identified and proceeded with Unit selection mechanism. The datasets used for this study were gathered from newspaper articles and the corresponding sentences were recorded by a professional speaker. User level evaluation was conducted with 20 candidates, where the intelligibility and the naturalness of the developed Sinhala TTS system received an approximate score of 70%. And the overall speech quality is an approximately to 60%.

Index Terms: Text to Speech, Unit Selection, Phonetics, Sinhala TTS, Sinhala Phonology

1. Introduction

A Text to Speech (TTS) can be recognized as the computer based system which is capable of converting text to its desired spoken form considering a grapheme to phoneme mapping. This application is useful in many industries such as Finance, Transportation, Medical and Entertainment. TTS systems are capable of reading text from different resources such as newspaper articles, websites or e-books, and convert those analyzed text into synthesized spoken language through a computational mechanism. This mechanism consists of 3 subsystems; text analysis, linguistic analysis and waveform generation [1]. Text analysis level consists of identifying and converting the non-textual content into a human recognizable text through tokenization and normalization process. In text normalization phase it will assign prosodic rules; intonation, phrasing, stress and pause into the identified string which wants to be read. Waveform generation involves the construction of an acoustic signal from desired spoken language by representing the synthesis approaches like formant synthesis, articulatory synthesis and waveform concatenation [2].

Through this paper, we propose an approach to implement a text to speech (TTS) system for Sinhala Language using a unit selection mechanism. For the implementation phase we selected MaryTTS framework which is considered as portable, user friendly, Java based open source framework which supports for multilingual languages [3]. Research have been carried out to implement TTS systems for German, Hindi, Telugu and Turkish languages using this framework and the same mechanism, and they have received acceptable results [4].

The structure of this paper is as follows; in Section 2 we describe the design and implementation process of the TTS. We describe the evaluation & results in Section 3. The paper concludes with the conclusion in Section 4 & acknowledgements in Section 5.

2. Methodology and Design

The proposed solution for building the Sinhala TTS has been made up with two major components; i.e preparation of training data set followed by feature extraction from the extracted text files based on the pre-defined letter to sound rules. Then the final component consists of building a new synthesis voice for Sinhala language.

When developing a new voice using MaryTTS[4], the framework automatically creates a set of sub files as a prerequest to the voice building phase. Therefore, we had to customize those modules according to Sinhala language requirements. Furthermore, language specified files (lexicon, MFCC statistical data extraction, tokenization, and allophone) and waveform specified files (duration according to the letter to sound rules and intonation) are extracted through the compilation process[1]. The entire voice creation process consists of 5 steps and organized as follows;

- Defining the set of allophones for Sinhala language.
- Letter to sound conversion.
- Wave file generation process.
- New voice compilation process.
- Voice integration into the Windows platform.

2.1. Defining the allophone set for Sinhala language

Sinhala is one of the official languages in Sri Lanka spoken by 74% of its population [5]. According to linguistic studies, spoken Sinhala contains 40 phonemes together with 14 vowels (Figure 1) and 26 consonants (Figure 2). However, in this work we were requested by the visually impaired community in Sri Lanka to consider all the graphemes in Sinhala alphabet as separate phonemes, for them to be able to identify all the graphemes by listening. Therefore aspirated / non-aspirated and retroflex / dental distinctions were also considered to make the set of allophones for the training, though these distinctions are only seen in language used in specific occasions; i.e. dharma sermon and professional announcing etc. Owing to lack of use in contemporary Sinhala, independent vowels; *iruuyanna* (0D8E), *iluyanna* (0D8F), *iluuyanna* (0D90) and dependent vowels; *diga gaettapilla* (0DF2), *gayanukitta* (0DDF) and *diga gayanukitta* (0DF3) were not considered in preparation the allophone set[6]. In addition to the independent vowels, there are two letters with their corresponding dependent vowels for diphthongs /ai/ and /au/ in Sinhala. The other diphthongs used in spoken Sinhala are /iu/, /eu/, /u/, /ou/, /ui/, /ei/, /i/, and /oi/ [7]. From the consonants, only palatal pre-nasalized voiced stop was not included in the allophone set.

10.21437/SLTU.2018-33

Independent Vowels	Dependent Vowels	Pronunciation
අ		a
ආ	ා	a:
ඇ	ැ	æ
ඈ	ෑ	æ:
ඉ	ි	i
ඊ	ී	i:
උ	ු	u
ඌ	ූ	u:
ඍ	ෘ	ri
Not Used	ෲ	ri:
එ	ෙ	e
ඒ	ේ	e:
ඓ	ෛ	ai
ඔ	ො	o
ඕ	ෝ	o:
ඖ	ෞ	au

Figure 1: *Sinhala vowel classification used for the TTS.*

			Lab.	Den.	Alv.	Ret.	Pal.	Vel.	Glo.
Stops	-Voi	-Asp	ප p	ත t		ට ʈ		ක k	
		+Asp	ඵ pʰ	ථ tʰ		ඨ ʈʰ		ඛ kʰ	
	+Voi	-Asp	බ b	ද d		ඩ ɖ		ග g	
		+Asp	භ bʰ	ධ dʰ		ඪ ɖʰ		ඝ gʰ	
Affricates	-Voi	-Asp					ච c		
		+Asp					ඡ cʰ		
	+Voi	-Asp					ජ ɟ		
		+Asp					ඣ ɟʰ		
Pre-nasalized voiced stops			ඹ mb	ඳ nd		ඬ ɳɖ		ඟ ŋg	
Nasals			ම m		න ණ n		ඤ ɲ	ඞ ŋ	
Trill						ර r			
Lateral					ල ළ l				
Spirants			ෆ f	ස s			ශ ෂ ʃ		හ h
Semivowels			ව v				ය j		

Figure 2: *Sinhala consonant classification used for the TTS.*

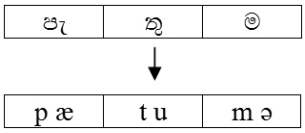

Figure 3: *G2P mapping for non-aspirated sounds.*

2.2. Letter to sound conversion

The letter to sound conversion phase is responsible of converting orthographic text into its corresponding phonetic representation based on the characteristics of desired language module. In Sinhala most of the letter to sound mappings are represented using one-to-one mapping in between letters and phoneme. To proceed with this section, the data set was retrieved from UCSC 10M Sinhala corpus. We gathered most frequent 5000 words to build the pronunciation dictionary. Building a pronunciation dictionary is a prominent task in developing a TTS system. Therefore, as the project progressed, initial lexicon was changed based on accuracy and made several versions as discussed in 2.4.

Since Sinhala is an abugida or alphasyllabary writing system, consonantvowel(CV) sequences are written as a unit. Given example in Figure 3 consists of six phones but has written in three letters with modifiers.

In the initial pronunciation lexicon non-aspirated consonants were represented in one phonetic letter but /h/ was used to denote aspiration. As in Figure 4, three phonetic representations /gha/ has used to denote one CV sequence. Several versions of the pronunciation dictionary were built as a result of re-iteration of the process by changing the training samples. Version 5 of the built TTS system contains the mapping of one phoneme to one character and is the final version of the pronunciation dictionary.

2.3. Wave file generation process

Generating wave file from a native speaker is a pre request of building a synthesis voice in MaryTTS framework. Therefore, the Redstart voice recording tool can be used for creation of a voice in the target language [8]. articles through UCSC Sinhala corpus. Moreover, we have to record those extracted sentences using a native speaker considering the set of characteristics such as pitch, period and noise. Then we have to convert those audio files to map with the MaryTTS requirements before building a synthesis voice from it. In MaryTTS architecture we converted those recorded wav files into 16 kHz sampling frequency and 16-bit sample format through a separate interface.

2.4. A new voice compilation process

One way to generate the synthesis voice from MaryTTS framework would be to study and understand the key features of each and every sub levels in the architecture. A new synthesis voice building phase supports unit selection and Hidden Markov module (HMM) mechanism and upon them our approach is implemented based on unit selection mechanism. A preliminary stage involves the recording of wave files using the Redstart tool and creating the corresponding text of what is spoken in each wav file in the .txt format.

This architecture breaks down to 9 dependent sub modules such as, feature extraction from acoustic data, support for transcription conversion, automated labeling, label transcription alignment, feature vector extraction from text data, verify alignment, basic data files, building acoustic models and finally creation of unit selection files. These sub modules automatically generate a set of intermediate output files and those files contains a set of values proceeding to the next sub module. This phase is responsible for generating a set of statistical files from

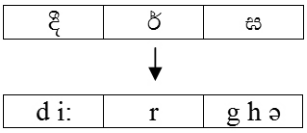

Figure 4: *G2P mapping for aspirated sounds.*

speech wave files such as MFCC calculation, together with the extracted allophones from the corresponding text of what is spoken in each speech signal. Furthermore, it will create label files which are responsible for producing an aligned label set while listening to each corresponding wave file. Label-transcription Alignment phase is used to create 3 sub directories to store converted Mary phone and Mary half-phone labels from previously generated label files. Furthermore, in this phase it will automatically create allophone files in xml format[4][9].

In our implementation process we used 1000 sentences in text format with their corresponding recorded wav files for training purpose and we maintained the same corpus throughout all our 5 versions. A brief description of each version is given below.

- **Version 1:** Built a synthesis Sinhala voice using a corresponding pronunciation dictionary. It consists of handling the aspiration and non-aspiration sounds. Here we used diphone mapping for handling non-aspiration sounds using two characters such that capital letter followed by a simple letter. We handled the aspiration sounds just using a simple letter. Moreover, voice integration phase consists of 1000 sentences with their corresponding wav files. However, through this voice it failed to differentiate between /a/ and /ə/ sounds. Therefore, we moved to our next version.
- **Version 2:** To resolve the problems that occurred in the previous version we used a single letter to represent the phonemes in Sinhala language. Here we used a combination of simple letters for handling non-aspiration sounds & capital letters for handling aspiration sounds. However, voice building phase was carried out using the same text data set that we used in previous version. Through this version it gave much better performance with compared to the previous version.
- **Version 3:** Both of our previous versions failed to produce a smooth voice as we expected. Therefore, we manually analyzed and changed allophone xml files, with the guidance of domain experts, which were generated from the 3rd sublevel in the voice building phase without changing the pronunciation dictionary and the text data that we used in our 2nd version. Upon that decision we didn't retrieve much better performance compared to the previous voices.
- **Version 4:** However, when we listened to the synthesis voice that we build from our latest version (3rd version) we encountered that there were some misalignments between wav files and their corresponding text files. Therefore we have analyzed and altered those lab files. For this task we used audacity which support in the Linux environment for audio analyzing. From that modification we retrieved a smooth Sinhala voice from our approach and it gives much better performances with compared to others.
- **Version 5:** For an experiment here we apply the combination of the concepts that we followed in both 3rd and 4th versions. Finally we observed that we didn't retrieve much improvement compared to the 4th version results. Therefore we moved with our evaluation phase by considering the 4th version.

2.5. Voice integration into the Windows platform

The MaryTTS framework is only compatible with Linux environments. For the voice integration process we have to deal with the screen reader. Among them we choose NVDA as our screen reader because most of the users are aware of Windows platform and NVDA is one of the most used screen reader in that platform combining with the speechhub which acts as a bridge between MaryTTS voice and NVDA screen reader.

NVDA (Non Visual Desktop Access) is known as an open source screen reader which allows blind and visually impaired people to use computers. This screen reader is capable of reading the text on the screen in a defined computerized voice. User has an ability to control what is read by moving the cursor to the relevant area of text using a mouse or some other equipment [10].

3. Evaluation & Results

The text to speech system can be evaluated by comparing the computerized voice and human voice, considering the various aspects such as naturalness, intangibility and suitability for used application [11]. Sinhala is generally considered as morphologically rich language based on their characteristics. Therefore, among them the most important components in our evaluation phase are the intelligibility and the naturalness of the speech.

The intelligibility of the speech refers that the synthesizer's output voice could be understood by a native person while the naturalness of the speech refers that the computerized spoken voice is similar to the human voice[12]. Specially, the TTS system is intended to use in various application; transportation, entertainment and medical by visually impaired people. Therefore, the naturalness scoring method proposed by Sluijter et al. estimates eleven factors which each listener is asked to map with the values from five-point scale [13].

After integrating the voice with the NVDA, Beta-testing for first five versions of the TTS was carried out using 20 samples among impaired people and another ten among sited people, to avoid the biasness. Here we used overall 15 sentences from 3 categories; 5 sentences with words from the training dataset, 5 sentences with words which are not in the training dataset and 5 sentences from editorial articles of online newspapers.

For this evaluation process we recorded the voice generated from the build Sinhala TTS for those 15 selected sentences and then we facilitated to listen those pre-recorded 15 sentences for each individual in our testing sample and asked them to write down the sentence they can hear while ranking the speech quality and the naturalness of them according to the grey scale given. The responses were marked separately in the evaluation sheets and the analysis were made based on those responses. Based on the observation of re-written sentences the intelligibility of the Sinhala TTS system was measured, which was defined as follows (in 1);

$$Intelligibililty = Avarage(\sum_{n=1}^{20}(100\frac{X}{Y})) \qquad (1)$$

Where X = number of correctly identified words and Y = total number of words in the sentences

Based on the results from those two samples we calculate the quality of the TTS for both visually impaired (shown in Figure 5) and for the sited samples (shown in Figure 6) and finally the overall performance was calculated.

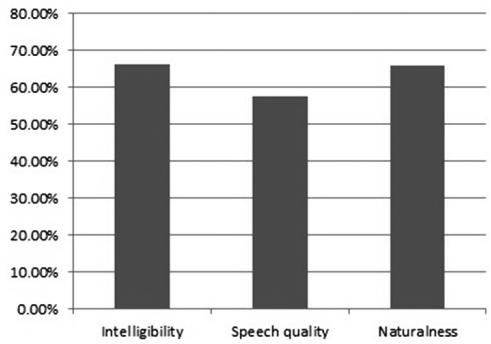

Figure 5: *Quality of the synthesis Sinhala voice in visually impaired category.*

According to figure 5, intangibility of the Sinhala TTS is 66% based on the results we received from the visually impaired category. It means 66 words out of 100 pronounced by the Sinhala TTS can be correctly identified for the visually impaired people. Speech quality of the TTS is also above 50% and it indicate the improvement of usability aspect with respect to the previous TTS system; Festival which is implemented by LTRL [1].

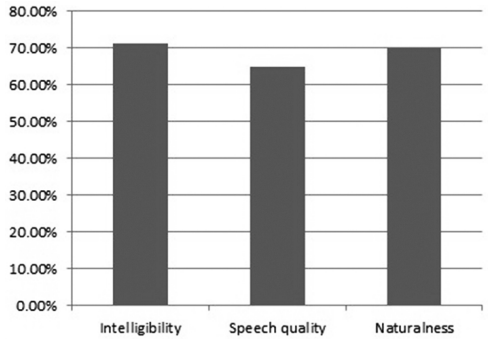

Figure 6: *Quality of the synthesis Sinhala voice in sited category.*

Further, sited category has given higher values for the speech quality and the naturalness (70%) as well, may be due to the same reason. Since the sited category have no previous experience on listening TTS voices, they may have heard the Sinhala TTS voice better than the visually impaired evaluators. Further we calculated the overall performance of the Sinhala TTS by combining the values of both visually impaired and sited evaluators and Figure 7 below shows the values assigned for each attributes. According to figure 7 below, it clearly depicts that the intelligibility of the developed Sinhala TTS system received an approximately to 70%. Naturalness of the voice is also just below 70% while the overall speech quality is above 60%.

We identified two types of errors in the error analysis. The first type of errors includes phones not properly read. For instance /a/ and / ə / in Sinhala are two allophones in one phoneme and they occurred frequently in Sinhala speech. The TTS not properly read these two, since randomly selected 1000 sentences are not enough to cover most of context dependent sound units. The second type of errors includes digits, abbreviations and other such signs which are left in reading. To treat them a normalization process need to be applied. We hope implementing them in the future.

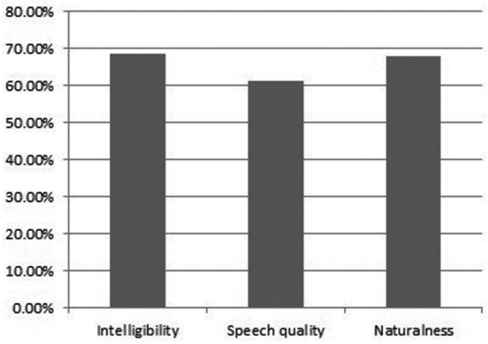

Figure 7: *Overall quality.*

4. Conclusion

In order to record the used sentence corpus, we hired a female announcer. Even though aspirated consonants are pronounced non-aspirated in colloquial Sinhala, they are sometimes pronounced in formal speaking. Based on that consideration of that variation and recorded sentences with aspirated consonants. This approach is aimed at developing the synthesis Sinhala voice to convert text to speech using a unit selection mechanism in MaryTTS framework. In Sri Lanka, majority of people are confident in interacting in Sinhala language which is known as morphologically rich language. Therefore, implementing a TTS system for Sinhala is a challenging task. Through our system we reduce the gap between information retrieval in visually impaired by considering the native Sinhala language. As planned initially, this approach succeeded in identifying and reading the corresponding Sinhala text based on the unit selection methodology. On perceiving the results gained through the evaluation phase it clearly stated this proposed approach gives above 50% of accuracy for visually impaired category and 70% accuracy for sited category. Through this research we improved the performance of the synthesis voice depend on this data set. Therefore, we hope to improve our pronunciation dictionary and text sentence corpus in our future improvements.

5. Acknowledgements

This work was funded by the World Intellectual Property Organization, Geneva 20, and Switzerland. We are grateful Mr. Asoka Bandula from DAISY Lanka Foundation for his invaluable support in this project. We also wish to acknowledge Miss Sumudu Nanayakkara for providing voice for the TTS and the students of the Faculty of Arts, University of Colombo who supported the evaluation. The authors also acknowledge all the members of Language Technology Research Laboratory of the University of Colombo School of Computing, Sri Lanka, who helped in various ways to make this work bear fruit.

6. References

[1] R. Weerasinghe, A. Wasala, V. Welgama, and K. Gamage, "Festival-si: A sinhala text-to-speech system," in *International Conference on Text, Speech and Dialogue*. Springer, 2007, pp. 472–479.

[2] R. Weerasinghe, A. Wasala, and K. Gamage, "A rule based syllabification algorithm for sinhala," in *International Conference on Natural Language Processing*. Springer, 2005, pp. 438–449.

[3] M. Schröder and A. Hunecke, "Creating german unit selection voices for the mary tts platform from the bits corpora," *Proc. SSW6, Bonn, Germany*, 2007.

[4] S. Pammi, M. Charfuelan, and M. Schröder, "Multilingual voice creation toolkit for the mary tts platform." in *LREC*. Citeseer, 2010.

[5] R. N. Kearney, "Sinhalese nationalism and social conflict in ceylon," *Pacific Affairs*, vol. 37, no. 2, pp. 125–136, 1964.

[6] P. Daniels, "The unicode consortium: The unicode standard," *Language: journal of the Linguistic Society of America*, vol. 69, no. 1, pp. 225–225, 1993.

[7] A. Weerasinghe, R. Wasala, and K. Gamage, "Sinhala grapheme-to-phoneme conversion and rules for schwa epenthesis." Proceedings of the COLING/ACL Main Conference Poster Sessions, Association for Computational Linguistics, 2006.

[8] M. Schröder, M. Charfuelan, S. Pammi, and I. Steiner, "Open source voice creation toolkit for the mary tts platform," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[9] S. Le Maguer and I. Steiner, "Uprooting marytts: Agile processing and voicebuilding," in *28th Conference on Electronic Speech Signal Processing (ESSV), Saarbrücken, Germany*, 2017.

[10] N. Access, "Nvda," 2010.

[11] M. Karjalainen, "Review of speech synthesis technology," *Helsinki University of Technology, Department of Electrical and Communications Engineering*, 1999.

[12] M. Rashad, H. M. El-Bakry, and I. R. Isma'il, "Diphone speech synthesis system for arabic using mary tts," *International Journal of Computer Science & Information Technology*, vol. 2, no. 4, 2010.

[13] A. Sluijter, E. Bosgoed, J. Kerkhoff, E. Meier, T. Rietveld, A. Sanderman, M. Swerts, and J. Terken, "Evaluation of speech synthesis systems for dutch in telecommunication applications," in *The Third ESCA/COCOSDA Workshop (ETRW) on Speech Synthesis*, 1998.